Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for modifying a graph-based representation of an executable ~~computer~~ data processing application, the graph-based representation including a graph having vertices representing components and links between components indicating flows of data between such components, ~~the graph further having~~ at least some of the components comprising data processing components ~~with~~ having parameters, including:

programmatically retrieving definitions of runtime parameters for the graph at runtime execution of the graph, the runtime parameters being defined as determinable at runtime execution of the graph;

determining whether a value for each of the runtime parameters is to be provided by user input;

determining whether a value for each of the runtime parameters is to be externally supplied programmatically;

displaying a prompt to a user for receiving user input for every runtime parameter so determined to be provided by user input;

retrieving any externally supplied value for every runtime parameter determined to be externally supplied programmatically;

determining parameter values for the runtime parameters based on the user input to such prompt or such externally supplied value or a default value;

modifying the graph-based representation of the data processing application using the determined parameter values for the runtime parameters; and

executing the data processing application represented by the modified graph-based representation to process at least one flow of data received by at least one data processing component.

2. (Canceled)

3. (Original): The method of claim 1, further including providing an interface which permits designating a parameter of a graph component as a runtime parameter.

4. (Previously Presented): The method of claim 1, wherein determining the parameter values includes evaluating an expression.

5. (Original): The method of claim 4, wherein the expression computes metadata.

6-7. (Canceled)

8. (Previously presented): The method of claim 1, wherein a prompt for receiving user input is conditional, and displaying the prompt depends upon evaluation of user input to a prior displayed prompt.

9. (Currently amended): A method for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the method including:

~~(a)~~ determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having an associated condition and an associated condition-interpretation;

~~(b)~~ evaluating the associated condition for every such conditional component;

~~(c)~~ modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by removing such conditional component and all connected flows to such conditional component from the graph before execution of the graph, based on an evaluation of the associated condition and the corresponding associated condition-interpretation for such conditional component; and

~~(d)~~ executing the application represented by the modified graph.

10. (Canceled)

11. (Previously presented): The method of claim 9, further including removing each component and flows connected to such components that depend on the presence of the removed conditional component.

12. (Currently amended): A method for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the method including:

~~(a)~~ determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having [[a]] an associated condition and an associated condition-interpretation;

~~(b)~~ evaluating the associated condition for every such conditional component;

~~(c)~~ modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by replacing such conditional component with a flow before execution of the graph based on an evaluation of the associated condition and the corresponding condition-interpretation for such conditional component; and

~~(d)~~ executing the application represented by the modified graph.

13. (Previously presented): The method of claims 9 or 12, further including providing an interface which permits designating a condition and a condition-interpretation for a graph component.

14. (Currently Amended): A system for modifying a graph-based representation of an executable ~~computer~~ data processing application, the graph-based representation including a graph having vertices representing components and links between components indicating flows of data between such components, ~~the graph further having~~ at least some of the components comprising data processing components ~~with~~ having parameters, including:

means for programmatically retrieving definitions of runtime parameters for the graph at runtime execution of the graph, the runtime parameters being defined as determinable at runtime execution of the graph;

means for determining whether a value for each of the runtime parameters is to be provided by user input;

means for determining whether a value for each of the runtime parameters is to be externally supplied programmatically;

means for displaying a prompt to a user for receiving user input for every runtime parameter so determined to be provided by user input;

means for retrieving any externally supplied value for every runtime parameter determined to be externally supplied programmatically;

means for determining parameter values for the runtime parameters based on the user input to such prompt or such externally supplied value or a default value;

means for modifying the graph-based representation of the data processing application using the determined parameter values for the runtime parameters; and

means for executing the data processing application represented by the modified graph-based representation to process at least one flow of data received by at least one data processing component.

15. (Canceled)

16. (Original): The system of claim 14, further including an interface which permits designating a parameter of a graph component as a runtime parameter.

17. (Previously Presented): The system of claim 14, wherein the means for determining the parameter values includes means for evaluating an expression.

18. (Original): The system of claim 17, wherein the expression computes metadata.

19-20. (Canceled)

21. (Previously presented): The system of claim 14, wherein a prompt for receiving user input is conditional, and displaying the prompt depends upon evaluation of user input to a prior displayed prompt.

22. (Currently amended): A system for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the system including:

~~(a)~~ means for determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having an associated condition and an associated condition-interpretation;

~~(b)~~ means for evaluating the associated condition for every such conditional component;

~~(c)~~ means for modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one

such conditional component by removing such conditional component and all connected flows to such conditional component from the graph before execution of the graph, based on an evaluation of the associated condition and the corresponding associated condition-interpretation for such conditional component; and

~~(d)~~ means for executing the application represented by the modified graph.

23. (Canceled)

24. (Previously presented): The system of claim 22, further including means for removing each component and flows connected to such components that depend on the presence of the removed conditional component.

25. (Currently amended): A system for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the system including:

~~(a)~~ means for determining at runtime execution of the graph whether any component of the graph is defined as being a conditional component having [[a]] an associated condition and an associated condition-interpretation;

~~(b)~~ means for evaluating the associated condition for every such conditional component;

~~(c)~~ means for modifying the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by replacing such conditional component with a flow before execution of the graph based on an evaluation of the associated condition and the corresponding associated condition-interpretation for such conditional component; and

~~(d)~~ means for executing the application represented by the modified graph.

26. (Previously presented): The system of claims 22 or 25, further including an interface which permits designating a condition and a condition-interpretation for a graph component.

27. (Currently Amended): A computer program, stored on a computer-readable medium, for modifying a graph-based representation of an executable ~~computer~~ data processing application, the graph-based representation including a graph having vertices representing components and links between components indicating flows of data between such components, ~~the graph further having~~ at least some of the components comprising data processing components ~~with~~ having parameters, the computer program comprising instructions for causing a computer to:

programmatically retrieve definitions of runtime parameters for the graph at runtime execution of the graph, the runtime parameters being defined as determinable at runtime execution of the graph;

determine whether a value for each of the runtime parameters is to be provided by user input;

determine whether a value for each of the runtime parameters is to be externally supplied programmatically;

display a prompt to a user for receiving user input for every runtime parameter so determined to be provided by user input;

retrieve any externally supplied value for every runtime parameter determined to be externally supplied programmatically;

determine parameter values for the runtime parameters based on the user input to such prompt or such externally supplied value or a default value;

modify the graph-based representation of the data processing application using the determined parameter values for the runtime parameters; and

execute the data processing application represented by the modified graph-based representation to process at least one flow of data received by at least one data processing component.

28. (Canceled)

29. (Original): The computer program of claim 27, further including instructions for causing the computer to provide an interface which permits designating a parameter of a graph component as a runtime parameter.

30. (Previously Presented): The computer program of claim 27, wherein the instructions for causing the computer to determine the parameter values include instructions for causing the computer to evaluating an expression.

31. (Original): The computer program of claim 30, wherein the expression computes metadata.

32-33. (Canceled)

34. (Previously presented): The computer program of claim 27, wherein a prompt for receiving user input is conditional, and displaying the prompt depends upon evaluation of user input to a prior displayed prompt.

35. (Currently amended): A computer program, stored on a computer-readable medium, for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the computer program comprising instructions for causing a computer to:

~~(a)~~ determine at runtime execution of the graph whether any component of the graph is defined as being a conditional component having an associated condition and an associated condition-interpretation;

~~(b)~~ evaluate the associated condition for every such conditional component;

~~(c)~~ modify the graph at runtime execution of the graph in accordance with such evaluation and the associated corresponding condition-interpretation of at least one such conditional component by removing such conditional component and all connected flows to such conditional component from the graph before execution of the graph, based on an evaluation of the associated condition and the corresponding associated condition-interpretation for such conditional component; and

~~(d)~~ execute the application represented by the modified graph.

36. (Canceled)

37. (Previously presented): The computer program of claim 35, further including instructions for causing the computer to remove each component and flows connected to such components that depend on the presence of the conditional component.

38. (Currently amended): A computer program, stored on a computer-readable medium, for modifying a graph at runtime execution of the graph, the graph representing an executable computer application and having vertices representing components with parameters and links between components indicating flows of data between such components, the computer program comprising instructions for causing a computer to:

~~(a)~~ determine at runtime execution of the graph whether any component of the graph is defined as being a conditional component having [[a]] an associated condition and an associated condition-interpretation;

~~(b)~~ evaluate the associated condition for every such conditional component;

~~(c)~~ modify the graph at runtime execution of the graph in accordance with such evaluation and the corresponding associated condition-interpretation of at least one such conditional component by replacing the conditional component with a flow before execution of

the graph based on an evaluation of the condition and the corresponding condition-interpretation for such conditional component; and

~~(d)~~ execute the application represented by the modified graph.

39. (Previously presented): The computer program of claims 35 or 38, further including instructions for causing the computer to provide an interface which permits designating a condition and a condition-interpretation for a graph component.

40. (New) The method of claim 1, wherein the flow of data is received by at least one data processing component from at least one dataset component.

41. (New) The system of claim 14, wherein the flow of data is received by at least one data processing component from at least one dataset component.

42. (New) The computer program of claim 27, wherein the flow of data is received by at least one data processing component from at least one dataset component.